



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/825,466	04/15/2004	Timothy Nephi Tillotson	10030536-1	8111

7590 06/20/2008
AGILENT TECHNOLOGIES, INC.
Legal Department, DL429
Intellectual Property Administration
P.O. Box 7599
Loveland, CO 80537-0599

EXAMINER

YU, HENRY W

ART UNIT	PAPER NUMBER
----------	--------------

2182

MAIL DATE	DELIVERY MODE
-----------	---------------

06/20/2008

PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No. 10/825,466	Applicant(s) TILLOTSON ET AL.	
	Examiner HENRY YU	Art Unit 2182	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) ☒ Responsive to communication(s) filed on 15 April 2004.
- 2a) ☐ This action is **FINAL**. 2b) ☒ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) ☒ Claim(s) 1-20 is/are pending in the application.
- 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 5) ☐ Claim(s) _____ is/are allowed.
- 6) ☒ Claim(s) 1-20 is/are rejected.
- 7) ☐ Claim(s) _____ is/are objected to.
- 8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) ☒ The specification is objected to by the Examiner.
- 10) ☒ The drawing(s) filed on 15 April 2004 is/are: a) ☒ accepted or b) ☐ objected to by the Examiner.
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some * c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
 2. ☐ Certified copies of the priority documents have been received in Application No. _____.
 3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- | | |
|--|---|
| 1) <input checked="" type="checkbox"/> Notice of References Cited (PTO-892) | 4) <input type="checkbox"/> Interview Summary (PTO-413) |
| 2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948) | Paper No(s)/Mail Date. _____ |
| 3) <input checked="" type="checkbox"/> Information Disclosure Statement(s) (PTO/SB/08) | 5) <input type="checkbox"/> Notice of Informal Patent Application |
| Paper No(s)/Mail Date <u>05/18/2004</u> . | 6) <input type="checkbox"/> Other: _____ |

DETAILED ACTION

The instant application having Application No. 10/825,466 has a total of 20 claims pending in the application; there are 3 independent claims and 17 dependent claims, all of which are ready for examination by the examiner.

INFORMATION CONCERNING DRAWINGS

Drawings

1. The applicant's drawings submitted are acceptable for examination purposes.

OBJECTIONS TO THE SPECIFICATION

Specification

2. The disclosure is objected to because of the following informalities:

On [Page 12, line 14], item 534 (disclosed as a ".NET event signal") is disclosed, though such an item is not mentioned in the drawings. Furthermore, there is no notation stating that such an item is not explicitly shown in the drawings (e.g. on page 8, items 128 and 129 are mentioned with the accompanying notation that both items are not explicitly shown in the drawings). Furthermore, the passage in which item 534 is disclosed strongly implies that the ".NET event signal" is instead referring to item 524.

Appropriate correction is required.

Claim Objections

3. **Claims 2, 4, 7, 9, 11, 14, and 20** are objected to because of the following informalities:

In **claim 2**, the word “*of*” should be placed between the words “*step*” and “*converting*” on lines 3 and 7. The same wording problem also applies to **claim 9** (lines 4 and 8).

In **claim 4**, Examiner suggests that Applicant place the word “*of*” between the words “*step*” and “*transferring*” on line 3. The same wording problem also applies to **claim 11** (line 4).

In **claim 7**, Examiner suggests that Applicant place the word “*of*” between the words “*step*” and “*asynchronously*” on line 3. The same wording problem also applies to **claim 14** (line 3).

In **claim 20**, Examiner suggests that Applicant place the word “*event*” before the word “*message*” on line 4, as the only previous mention of a “*message queue*” is an “*event message queue*” on line 3.

Appropriate correction is required.

REJECTIONS NOT BASED ON PRIOR ART

Claim Rejections - 35 USC § 112

4. The following is a quotation of the second paragraph of 35 U.S.C. 112:

The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

5. **Claims 1-14, 17-18, and 20** are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

Claim 1 recites the limitation “*the communication*” on lines 5, 14, and 23. It is unclear which communication is being referred to as two types of communications (SCPI protocol and .NET protocol communications) have been disclosed earlier in the claim on lines 1-3. Examiner suggests Applicant remove the word “*the*” that is before “*communication*” in each of the described instances so as to not signify that these passages are pointing to an earlier instance of “*communication*.” The same deficiency is also present in **claim 8**. **Claims 2-7 and 9-14** are also rejected as they inherit the deficiency that is present in parent **claims 1 and 8** respectively.

Claim 3 recites the limitation “*the query or the command*” on line 3. However, it is unclear which query or command is being referred to as there are two types disclosed in parent **claim 1** (SCPI and .NET). Examiner assumes, for the purpose of examination, that the type referred to be SCPI as the claim then discloses that a .NET response is formed. The same deficiency is also present in **claim 10**. **Claim 11** is also rejected as it inherits the deficiency that is present in parent **claim 10**.

Claim 8 recites the limitation “*the computer*” on line 2. There is insufficient antecedent basis for this limitation in the claim. Examiner suggests that Applicant replace the word “*the*” that is before “*computer*” with the word –*an*–.

Claim 17 recites the limitation “*the instrument application*” on line 3. There is insufficient antecedent basis for this limitation in the claim. Examiner suggests that Applicant replace the word “*the*” that is before “*instrument application*” with the word –*an*–. The same deficiency also applies to **claim 20**. **Claim 18** is also rejected as it inherits the deficiency that is present in parent **claim 10**.

REJECTIONS BASED ON PRIOR ART

Claim Rejections - 35 USC § 103

6. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

7. **Claims 1, 3-4, 8, 10-11, 15, and 16-18** are rejected under 35 U.S.C. 103(a) as being unpatentable over Fuller, III et al. (Patent Number US 7,134,081 B2) in view of Robison et al. (Publication Number US 2005/0060693 A1).

As per **claim 1**, Fuller, III et al. discloses an instrument controlling system that handles responses **[Abstract]**, as well as the use of SCPI **[Column 4, lines 55-57]** and .NET **[Column 5, lines 29-35]**. However, Fuller, III et al. does not explicitly disclose the idea of command translation in the limitations “*a method for translating between Standard Commands for Programmable Instrumentation (SCPI) protocol and .NET protocol communications, comprising: when the communication is a SCPI protocol command from a client, converting the SCPI protocol command to a .NET protocol command,*” “*evaluating the .NET protocol command to determine the validity of parameters sent from the client with the SCPI protocol command,*” “*otherwise, when the communication is a SCPI protocol query from the client, converting the SCPI protocol query to a .NET protocol query,*” “*evaluating the .NET protocol query to determine the validity of parameters sent from the client with the SCPI protocol query,*” or “*calling an*

appropriate Application Program Interface (API) of an instrument application, wherein the communication is intended for the instrument application and wherein the API is responsive to method calls in the .NET protocol.”

Robison et al. discloses the idea of command translation in the limitations “a method for translating between Standard Commands for Programmable Instrumentation (SCPI) protocol **(for this part of the rejection, labeled the 'first' protocol)** and .NET **(for this part of the rejection, labeled the 'second' protocol)** protocol communications, comprising: when the communication is a **(first)** protocol command from a client, converting the...protocol command to a **(second)** protocol command **(the command string is syntactically matched by the command processor code portion and all parameter values within the command string are converted to their corresponding data object; Page 2, paragraph 0022)**” and “evaluating the **(second)** protocol command to determine the validity of parameters sent from the client with the **(first)** protocol command **(the corresponding data objects can then be further validated; Page 2, paragraph 0022)**.” The same also applies to “otherwise, when the communication is a SCPI protocol query from the client, converting the SCPI protocol query to a .NET protocol query” and “evaluating the .NET protocol query to determine the validity of parameters sent from the client with the SCPI protocol query,” where a query can also act as a particular command subset that instructs a device/system to return a specific value (as noted by Fuller, III et al. **[Column 4, line 49]**).

Robison et al. discloses “calling an appropriate Application Program Interface (API) of an instrument application, wherein the communication is intended for the

instrument application and wherein the API is responsive to method calls in the...protocol (action handler code is invoked to actually execute the task that corresponds to the command; Page 2, paragraph 0022)."

Fuller, III et al. and Robison et al. are analogous art in that they are from the same field of command processing.

It would have been obvious to one of ordinary skill in the art at the time of invention to combine the method and system disclosed by Fuller, III et al. to include a means of translating and evaluating commands from one protocol to another as disclosed by Robison et al., which notes the need to create meaningful, actionable objected or data structures [Page 1, paragraph 0003]. Robison et al. also notes that there are prior art systems where the syntax of all commands is hard-coded into the parser [Page 1, paragraph 0004], and hence it would be necessary to translate commands protocols to accommodate such systems.

As per claims 3 and 10, the combination of Fuller, III et al. and Robison et al. discloses "the method" (see rejection to claims 1 and 8). Fuller, III et al. further discloses "when the query or the command is communication requiring response from the instrument application, forming a .NET protocol response message to the communication (a command is sent to the instrument (step 304), with a response from the instrument then being received (step 306); FIG. 9)."

While Robison et al. discloses the idea of command translation [Page 2, paragraph 0022], Fuller, III et al. discloses the handling of responses in the limitations "translating the (.NET) protocol response message to a SCPI protocol response

message (steps 306 to 310, where a response from the instrument is received, with the instrument response parsed and code created; FIG. 9), wherein the SCPI protocol response message comprises contents of nodes of a SCPI hierarchical tree structure (the instrument responses can be pre-parsed, which require the use of a library of data import functions to pre-parse instrument responses (Column 15, lines 22-25). It should be noted that the idea of a hierarchical tree is implied through Column 14, lines 47-52, where certain options may be only exposed when necessary, particularly through an 'Advanced Features' tab (which is a step beyond basic functions))" and "transferring the SCPI protocol response message to the client (a command is sent to the instrument (step 304), with a response from the instrument then being received (step 306); FIG. 9)." Claim 10 is rejected in a similar fashion.

As per claims 4 and 11, the combination of Fuller, III et al. and Robison et al. discloses "the method" (see rejection to claims 1 and 8). Robison et al. further discloses "before the method step transferring the (SCPI) protocol response message to the client, converting the (SCPI) protocol response message to (SCPI) format order (**all parameter values within the command string are to be converted to their corresponding data objects before any action handler code is invoked (Page 2, paragraph 0022). This passage demonstrates the need to convert data object types to the appropriate types before being processed (similar to the instant application's need to convert the protocol response message to the appropriate**

format before being sent to the client system)).” **Claim 11** is rejected in a similar fashion.

As per **claim 8**, Fuller, III et al. discloses “a computer readable memory device embodying a computer program of instructions executable by the computer (***a system and method for querying message-based instruments, automatically...parsing the responses, and generating code that encapsulates the connection/communication with the instrument and the parsing of the response; Column 4, lines 30-35***).” However, Fuller, III et al. does not disclose “the instructions comprising: when the communication is a SCPI protocol command from a client, converting the SCPI protocol command to a .NET protocol command,” “evaluating the .NET protocol command to determine the validity of parameters sent from the client with the SCPI protocol command,” “otherwise, when the communication is a SCPI protocol query from the client, converting the SCPI protocol query to a .NET protocol query,” “evaluating the .NET protocol query to determine the validity of parameters sent from the client with the SCPI protocol query,” or “calling an appropriate Application Program Interface (API) of an instrument application, wherein the communication is intended for the instrument application and wherein the API is responsive to method calls in the .NET protocol.”

Robison et al. discloses “the instructions comprising: when the communication is a SCPI (***for this rejection, the protocol is noted as a ‘first’ protocol***) protocol command from a client, converting the SCPI protocol command to a .NET (***for this rejection, the protocol is labeled as a ‘second’ protocol***) protocol command (***the***

command string is syntactically matched by the command processor code portion and all parameter values within the command string are converted to their corresponding data object; Page 2, paragraph 0022)” and “*evaluating the .NET (second) protocol command to determine the validity of parameters sent from the client with the SCPI (first) protocol command (the corresponding data objects can then be further validated; Page 2, paragraph 0022).*” The same also applies to “*otherwise, when the communication is a SCPI protocol query from the client, converting the SCPI protocol query to a .NET protocol query*” and “*evaluating the .NET protocol query to determine the validity of parameters sent from the client with the SCPI protocol query,*” where a query can also act as a particular command subset that instructs a device/system to return a specific values (as noted by Fuller, III et al. [Column 4, lines 49]). Robison et al. also discloses “*calling an appropriate Application Program Interface (API) of an instrument application, wherein the communication is intended for the instrument application and wherein the API is responsive to method calls in the .NET protocol (action handler code is invoked to actually execute the task that corresponding to the command; Page 2 paragraph 0022).*”

Fuller, III et al. and Robison et al. are analogous art in that they are from the same field of command processing.

It would have been obvious to one of ordinary skill in the art at the time of invention to combine the computer readable memory device embodying a computer program of instructions executable by the computer as disclosed by Fuller, III et al. to include a means of translating commands from one protocol to another as disclosed by

Robison et al., which notes the need to create meaningful, actionable objected or data structures **[Page 1, paragraph 0003]**. Robison et al. also notes that there are prior art systems where the syntax of all commands is hard-coded into the parser **[Page 1, paragraph 0004]**, and hence it would be necessary to translate commands protocols to accommodate such systems.

As per **claim 15**, Fuller, III et al. discloses “a system, comprising: a parser module configured to receive a Standard Commands for Programmable Instrumentation (SCPI) protocol communication from a client **(the system contains a means of parsing responses from an instrument (steps 308 to 310 in FIG. 9). Since the method is used to determine correct token characteristics (step 364 in FIG. 10), the same method can be applied in both directions to ensure proper command processing)** and to translate the SCPI protocol communication into a .NET protocol communication **(the result of the graphical token specification may be used to automatically create text-based code, such as .NET; Column 5, lines 29-31).**”

However, Fuller, III et al. does not explicitly disclose “an evaluator module, configured to evaluate the .NET protocol communication to determine the validity of parameters sent from the client with the SCPI protocol communication.”

Robison et al. discloses “an evaluator module **(command processor code portion)**, configured to evaluate the .NET protocol communication to determine the validity of parameters sent from the client with the SCPI protocol communication **(the corresponding data objects can then be further validated; Page 2, paragraph 0022).**” Robison et al. further discloses the idea of command translation in the

limitations “to translate the SCPI protocol communication into a .NET protocol communication **(the command string is syntactically matched by the command processor code portion and all parameter values within the command string are converted to their corresponding data object; Page 2, paragraph 0022).**”

Fuller, III et al. and Robison et al. are analogous art in that they are from the same field of command processing.

It would have been obvious to one of ordinary skill in the art at the time of invention to combine the system as disclosed by Fuller, III et al. to include a means of translating commands from one protocol to another as disclosed by Robison et al., which notes the need to create meaningful, actionable objected or data structures **[Page 1, paragraph 0003]**. Robison et al. also notes that there are prior art systems where the syntax of all commands is hard-coded into the parser **[Page 1, paragraph 0004]**, and hence it would be necessary to translate commands protocols to accommodate such systems.

As per **claim 16**, the combination of Fuller, III et al. and Robison et al. discloses “the system” (see rejection to **claim 15** above). While Fuller, III et al. discloses the use of SCPI **[Column 4, lines 55-57]** and .NET **[Column 5, lines 29-35]**, Robison et al. further discloses “a first format converter module configured to convert the SCPI protocol communication into a .NET stream format **(the command string is syntactically matched by the command processor code portion and all parameter values within the command string are converted to their corresponding data object; Page 2, paragraph 0022).**”

As per **claim 17**, the combination of Fuller, III et al. and Robison et al. discloses “the system” (see rejection to **claim 15** above). While Robison et al. discloses the idea of translation as **[the command string is syntactically matched by the command processor code portion and all parameter values within the command string are converted to their corresponding data object (Page 2, paragraph 0022)]**, Fuller, III et al. discloses handling responses as noted in the limitations “a first translator module configured to translate a .NET response from the instrument application to a SCPI protocol response (***the system receives a response from the instrument and parses the instrument response; FIG. 9; Column 15, lines 1-7; Column 20, lines 30-36***).”

As per **claim 18**, the combination of Fuller, III et al. and Robison et al. discloses “the system” (see rejection to **claim 15** above). While Robison et al. discloses the idea of translation as **[the command string is syntactically matched by the command processor code portion and all parameter values within the command string are converted to their corresponding data object (Page 2, paragraph 0022)]**, Fuller, III et al. discloses handling responses as noted in the limitations discloses “a second format converter module configured to convert the SCPI protocol response in a .NET stream format into SCPI format order (***the system receives a response from the instrument and parses the instrument response; FIG. 9; Column 15, lines 1-7; Column 20, lines 30-36***).”

8. **Claims 2 and 9** are rejected under 35 U.S.C. 103(a) as being unpatentable over Fuller, III et al. (Patent Number US 7,134,081 B2) in view of Robison et al. (Publication

Number US 2005/0060693 A1) and in further view of Durian et al. (Publication Number US 2002/0025832 A1).

As per **claims 2 and 9**, the combination of Fuller, III et al. and Robison et al. discloses “*the method*” (see rejection to **claims 1 and 8** above). However, the combination of Fuller, III et al. and Robison et al. does not explicitly disclose “*before the method step converting the SCPI protocol command to the .NET protocol command, placing the SCPI protocol command into .NET stream format*” or “*before the method step converting the SCPI protocol query to the .NET protocol query, placing the SCPI protocol command into .NET stream format.*”

Durian et al. discloses “*before the method step converting the SCPI protocol command to the .NET protocol command (**translating communications channel control commands within a wireless communication system**), placing the SCPI protocol command into .NET stream format (**communications are generally translated into a format required by the receiving device (in this case a telephone); Page 17, paragraph 0121**).*” Durian et al. also discloses “*before the method step converting the SCPI protocol query to the .NET protocol query, placing the SCPI protocol command into .NET stream format (**there is a situation where one type of formatting is done before another. In this case, communications channel control commands are translated into at least a first command selected from a set of system commands before being translated into corresponding wireless communication device control commands that can be understood by the receiving device; Pages 17-18, paragraph 0121**).*”

Fuller, III et al., Robison et al., and Durian et al. are analogous art in that they are from the same field of command processing.

It would have been obvious to one of ordinary skill in the art at the time of invention to combine the method/system as disclosed by the combination of Fuller, III et al. and Robison et al. to include as translation into a communication stream disclosed by Durian et al., which notes the desirability of having a plurality of different devices or applications to interconnect with a communications device as well as to communicate over a channel established by the by the communications device at substantially the same time **[Pages 1-2, paragraph 0010]**. **Claim 9** is rejected in similar fashion.

9. **Claims 5-7, 12-13, and 19** are rejected under 35 U.S.C. 103(a) as being unpatentable over Fuller, III et al. (Patent Number US 7,134,081 B2) in view of Robison et al. (Publication Number US 2005/0060693 A1) and in further view of Hall et al. (Patent Number US 5,974,541).

As per **claims 5 and 12**, the combination of Fuller, III et al. and Robison et al. discloses “*the method*” (see rejection to **claims 1 and 8** above). Though Fuller, III et al. discloses the use of .NET **[Column 5, lines 29-35]** and Robison et al. discloses the idea of command translation **[the command string is syntactically matched by the command processor code portion and all parameter values within the command string are converted to their corresponding data object; Page 2, paragraph 0022]**, which applies to the limitation “*converting the out of band signal IEEE 488.1 protocol signal to a .NET event,*” the combination of Fuller, III et al. and Robison et al. does not explicitly disclose “*asynchronously receiving an out of band IEEE 488.1 protocol signal*

from the client,” “converting the out of band signal IEEE 488.1 protocol signal to a .NET event,” or “transferring the out of band signal IEEE 488.1 protocol signal to the instrument application.”

Hall et al. discloses the use of IEEE 488.1, which applies to the limitation “converting the out of band signal IEEE 488.1 protocol signal **(the system utilizes GPIB, also known as the IEEE 388.1-1987; Column 1, lines 66-67)** to a .NET event.”

Hall et al. also discloses “asynchronously receiving an out of band IEEE 488.1 **(the system utilizes GPIB, also known as the IEEE 488.1-1987; Column 1, lines 66-67)** protocol signal from the client **(receiving a notify request from the GPIB application (step 302); FIG. 3)**” and “transferring the out of band signal IEEE 488.1 protocol signal to the instrument application **(anything sent from the GPIB application ends up at the GPIB hardware (FIG. 2). The process can be done asynchronously; Column 2, lines 38-40).**”

Fuller, III et al., Robison et al., and Hall et al. are analogous art in that they are from the same field of command processing.

It would have been obvious to one of ordinary skill in the art at the time of invention to modify the method disclosed by the combination of Fuller, III et al. and Robison et al. to include a means of processing IEEE 488.1 asynchronously as disclosed by Hall et al., which notes the desirability of asynchronous processing compared to prior art systems, where the GPIB driver level software has traditionally been required to poll a device to determine when an event occurs. However, such a process typically requires a large amount of unnecessary processor time, thus

consuming valuable CPU resources [**Column 2, lines 29-35**]. **Claim 12** is rejected in a similar fashion.

As per **claims 6 and 13**, the combination of Fuller, III et al. and Robison et al. discloses “*the method*” (see rejection to **claims 1 and 8** above). However, the combination of Fuller, III et al. and Robison et al. does not disclose “*when an event occurs in the instrument application, posting a notice of event occurrence in a status module*” or “*asynchronously notifying the client of event occurrence.*”

Hall et al. discloses “*when an event occurs in the instrument application, posting a notice of event occurrence in a status module (the method operates to asynchronously notify a user when one or more GPIB events occur in the GPIB system; Column 2, lines 40-42)*” and “*asynchronously notifying the client of event occurrence (represented by step 302 of receiving a notify request from the GPIB application; FIG. 3).*”

Fuller, III et al., Robison et al., and Hall et al. are analogous art in that they are from the same field of command processing.

It would have been obvious to one of ordinary skill in the art at the time of invention to modify the method as disclosed by the combination of Fuller, III et al. and Robison et al. to include a means of processing IEEE 488.1 asynchronously as disclosed by Hall et al., which notes the desirability of asynchronous processing compared to prior art systems, where the GPIB driver level software has traditionally been required to poll a device to determine when an event occurs. However, such a process typically requires a large amount of unnecessary processor time, thus

consuming valuable CPU resources [Column 2, lines 29-35]. **Claim 13** is rejected in similar fashion.

As per **claim 7**, the combination of Fuller, III et al., Robison et al., and Hall et al. discloses "the method" (see rejection to **claim 6 and 13** above). Fuller, III et al. further discloses "forming a .NET protocol response message to the query (**a command is sent to the instrument (step 304), with a response from the instrument then being received (step 306); FIG. 9**)" while Robison et al. further discloses "translating the .NET protocol response message to a SCPI protocol response message (**the command string is syntactically matched by the command processor code portion and all parameter values within the command string are converted to their corresponding data object; Page 2, paragraph 0022**)" and "transferring the SCPI protocol response message to the client (**all parameter values within the command string are to be converted to their corresponding data objects before any action handler code is invoked; Page 2, paragraph 0022**)."

Hall et al. further discloses "after the step asynchronously notifying the client of event occurrence (**represented by step 302 of receiving a notify request from the GPIB application; FIG. 3**), receiving query from the client requesting detailed information regarding the event occurrence (**the system has a means of ascertaining more information regarding the event occurrence including: monitoring events specified by the event information (step 304) and determining that an event specified by the information has occurred (step 306) before invoking a callback**

function in response to the event (step 308); FIG. 3). **Claim 14** is rejected in similar fashion.

As per **claim 19**, the combination of Fuller, III et al. and Robison et al. discloses “the system” (see rejection to **claim 15** above). Though Robison et al. discloses the idea of command translation as disclosed in “a third format converter module configured to convert an out of band IEEE 488.1 signal into a .NET signal (***the command string is syntactically matched by the command processor code portion and all parameter values within the command string are converted to their corresponding data object; Page 2, paragraph 0022***),” the combination of Fuller, III et al. and Robison et al. does not explicitly disclose “an out of band IEEE 488.1 signal,” which is disclosed by Hall et al. as **[the system utilizes GPIB, also known as the IEEE 488.1-1987 (Column 1, lines 66-67)]** and **[anything sent from the GPIB application ends up at the GPIB hardware (FIG. 2). The process can be done asynchronously (Column 2, lines 38-40)]**.

Fuller, III et al., Robison et al., and Hall et al. are analogous art in that they are from the same field of command processing.

It would have been obvious to one of ordinary skill in the art at the time of invention to modify the system as disclosed by the combination of Fuller, III et al. and Robison et al. to include a means of processing IEEE 488.1 asynchronously as disclosed by Hall et al., which notes the desirability of asynchronous processing compared to prior art systems, where the GPIB driver level software has traditionally been required to poll a device to determine when an event occurs. However, such a

process typically requires a large amount of unnecessary processor time, thus consuming valuable CPU resources [**Column 2, lines 29-35**].

10. **Claim 20** is rejected under 35 U.S.C. 103(a) as being unpatentable over Fuller, III et al. (Patent Number US 7,134,081 B2) in view of Robison et al. (Publication Number US 2005/0060693 A1) and in further view of Dobson et al. (Patent Number US 6,766,386 B2).

As per **claim 20**, the combination of Fuller, III et al. and Robison et al. discloses “the system” (see rejection to **claim 15** above). Though Robison et al. discloses the idea of command translation as disclosed in “an event translator module configured to receive notice of event occurrence from the status module and to translate that notice into a SCPI status notification (**the command string is syntactically matched by the command processor code portion and all parameter values within the command string are converted to their corresponding data object; Page 2, paragraph 0022**),” the combination of Fuller, III et al. and Robison et al. does not explicitly disclose “a status module comprising an event message queue and a status register wherein the message queue and the status register store event occurrence information from the instrument application.”

Dobson et al. discloses “a status module comprising an event message queue (**FIFO write control 414 in conjunction with a read data memory 416 in a first-in-first-out fashion; Column 6, lines 65-67; Column 7, lines 1-5**) and a status register (**status register 418; FIG. 4**) wherein the message queue and the status register store event occurrence information from the instrument application (**Column 8, lines 25-37**).”

Fuller, III et al., Robison et al., and Dobson et al. are analogous art in that they are from the same field of command processing.

It would have been obvious to one of ordinary skill in the art at the time of invention to modify the method as disclosed by the combination of Fuller, III et al. and Robison et al. to include a FIFO memory and status register as disclosed by Dobson et al. in order to prevent potential latencies and delays [**Column 1, lines 18-24**].

RELEVANT ART CITED BY THE EXAMINER

11. The following prior art made of record and relied upon is cited to establish the level of skill in the applicant's art and those arts considered reasonably pertinent to applicant's disclosure. See **MPEP 707.05(c)**.

12. The following references teach command processing and translation:

U.S. PATENT NUMBERS:

5,889,992

CLOSING COMMENTS

Conclusion

a. STATUS OF CLAIMS IN THE APPLICATION

13. The following is a summary of the treatment and status of all claims in the application as recommended by **M.P.E.P 707.07(i)**:

a(1). CLAIMS REJECTED IN THE APPLICATION

14. Per the instant office action, claims 1-20 have received a first action on the merits and are subject of a first action non-final.

15. The examiner requests, in response to this Office action, support be shown for language added to any original claims on amendment and any new claims. That is, indicate support for newly added claim language by specifically pointing to page(s) and line no(s) in the specification and/or drawing figure(s). This will assist the examiner in prosecuting the application.

16. Any inquiry concerning this communication or earlier communications from the examiner should be directed to HENRY YU whose telephone number is (571)272-9779. The examiner can normally be reached on Monday to Friday, 8:00 AM to 5:30 PM EST.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, TARIQ HAFIZ can be reached on (571) 272-6729. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/H. Y./
Examiner, Art Unit 2182

Application/Control Number: 10/825,466
Art Unit: 2182

Page 23

/Tariq Hafiz/
Supervisory Patent Examiner, Art Unit 2182